
MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
LIBRARIES

WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

An Accounting Model-Based Approach to Semantic Reconciliation in Heterogeneous Database Systems

Y. Richard Wang

October 1990 WP # MSA 3212-90

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

An Accounting Model-Based Approach to Semantic Reconciliation in Heterogeneous Database Systems

Y. Richard Wang

October 1990 WP # MSA 3212-90

Composite Information Systems Laboratory
E53-320, Sloan School of Management
Massachusetts Institute of Technology
Cambridge, Mass. 02139

Tel. (617) 253-0442
Fax (617) 492-4655
E-mail: rwang@sloan.mit.edu

ACKNOWLEDGEMENTS Work reported herein has been supported, in part, by MIT's International Financial Service Research Center and MIT's Center for Information Systems Research. The author wishes to thank Allen Moulton for his insight during the early stage of this research and Sung Khang for his field work.

An Accounting Model-Based Approach to Semantic Reconciliation in Heterogeneous Database Systems

ABSTRACT

Many important Management Support Systems require access to and seamless integration of multiple heterogeneous database systems. In this paper, we present a model-based approach for reconciling semantic heterogeneity of financial data retrieved from various data sources -- financial data which are needed in many Management Support Systems.

Specifically, a top-down, accounting model-based approach is presented to reconcile semantic heterogeneity among the data retrieved from multiple financial accounting databases. Knowledge represented in this model is applied to reconcile data conflicts and to infer new information. Accounting model-based rules for reconciling semantic heterogeneity at the data level are illustrated with intriguing cases using real data. To the best of our knowledge, this approach has not been applied to the heterogeneous database systems area for reconciling semantic heterogeneity at the data level as well as the schema level.

The accounting model-based approach checks the reliability and validity of data using knowledge encoded in the model. It enables Management Support Systems to produce integrated financial statements so that their users can focus on utilizing these integrated financial statements for tasks, such as profitability analysis, that concern them most.

An Accounting Model-Based Approach to Semantic Reconciliation in Heterogeneous Database Systems

I. Introduction

The increasing complexity, interdependence, and competition in the global business environment has profoundly changed how corporations operate and how they align their information technology for competitive advantage in the marketplace. It has been argued that the changing corporate operations will accelerate demands for more effective Management Support Systems for product development, product delivery, and customer service and management (Rockart & Short, 1989). With the increasing exploitation of database and telecommunication technologies in organizations and the dissemination of Management Support Systems from the executive to the line level, it is inevitable that many future systems will require dynamic access to information stored in disparate databases with disparate qualities, located both within and across organizational boundaries (Wang & Madnick, 1988).

Already, corporations are placing increasing emphasis on the management of data. Case studies of 31 data management efforts in 20 diverse firms have been reported (Goodhue, Quillard, & Rockart, 1988) in which five data management systems were identified:

(1) *Subject area databases* for operational systems containing data organized around important business entities or subject areas, such as customer and product.

(2) *Common systems* which are applications developed by a single, most often a central, organization to be used by multiple organizational units. For example, manufacturing applications such as production scheduling and spare parts inventory.

(3) *Information databases* which periodically draw their contents from operational databases and external sources, and often store data in aggregated forms.

(4) *Data access services* which focus mainly on improving managerial access to existing data, without attempting to upgrade the quality or structure of the data.

(5) *Architectural foundations* which are policies that force systems development efforts to conform to a well structured, overall plan.

The first three systems emphasize developing new databases or files with pertinent, accurate,

and consistent data. They require a major system development and on-going maintenance. Subject area databases and common systems are developed in firms seeking *better operational coordination;* whereas information databases are developed in firms seeking *improved managerial information.*

Data access services, on the other hand, are usually provided by a small group of personnel, often part of an information center, whose goal is to better understand what data is available in current systems and to put into place mechanisms to deliver this data. These mechanisms include locating appropriate data, extracting data from production files, or training users in fourth generation languages.

From the technical point of view, these systems represent two ends of a spectrum of approaches aiming at retrieving information from various data sources. At one end, data access services provide a "quick and dirty" approach for managers to get their hands on existing operational databases in a non-intrusive manner, thus maintaining the *local autonomy* of operational databases. At the other end, subject area databases, common systems, and information databases provide a *binding integration* of information by developing brand new databases, often at significant cost.

Evolution is an often-neglected factor in the development and deployment of these systems. Each system, as well as the needs for sharing among systems, will change over time. The rate and form of this evolution may tip the balance between autonomy and integration. Although autonomy and integration are conflicting factors, with evolution as a further complication, it may be possible to define a system architecture with sufficient flexibility to accommodate diverse requirements such as integration, autonomy, and evolution.

It is interesting to note, at this point, that database researchers have been actively addressing issues in developing such system architectures. They have referred to this type of system architectures as *heterogeneous database systems*[1], *Federated Database Systems* (Czejdo, Rusinkiewicz, & Embley, 1987; Elmasri, Larson, & Navathe, 1987; Heimbigner & McLeod, 1985; Lyngbaek & McLeod, 1983),

---

1 For example, National Science Foundation (NSF) sponsored a workshop on Heterogeneous Database Systems in cooperation with Northwestern University and IEEE-CS Technical Committee on Distributed Processing in December, 1989.

Multidatabases (Ferrier & Strangret, 1982; Litwin & Abdellatif, 1986; Litwin, et al., 1982)[2], or *Composite Information Systems* (Madnick, Siegel, & Wang, 1990; Wang & Madnick, 1988). The goal of heterogeneous database systems research is to cover a spectrum of capabilities in a coherent manner. These capabilities range from providing real-time connection to operational databases, to better operational coordination, to improved managerial information.

Many critical research and commercialization problems need to be reconciled in order to provide a solution for access to and seamless integration of heterogeneous database systems (Reiner, 1990). These problems include semantic reconciliation, data attribute tagging, networking, specification and processing of multidatabase queries, query optimization, transaction management, and tools for building multidatabases. In this paper, we focus on semantic reconciliation.

Reconciling semantic heterogeneity is an advanced issue in the heterogeneous database systems area. In the commercial world, establishing a physical network infrastructure and data access services have been the key concerns of executives making information systems decisions (Madnick & Wang, 1986). In a recent study of corporate needs for heterogeneous database systems[3], 80% of the managers surveyed cited data access as their immediate need. Speed of data access was cited as the second priority after data access was satisfied. The third priority was functionalities such as semantic reconciliation. Currently, semantic reconciliation is largely done by users. However, as users become more sophisticated, their attention will be directed to reconciliation of data values retrieved from disparate databases.

Researchers have started to address the reconciliation of data values through techniques such as frames and abstract data types (ADT). In the KSYS research project (Wiederhold, 1987), for example, the notion of frames as a data model was introduced. It investigates the implementation of a frame-based system residing at a level above the database schema to provide virtual attribute facility. In the CISL research project (Madnick, Siegel, & Wang, 1990), ADT was used to determine if

---

2 NSF will sponsor a similar workshop on Multidatabases & Semantic Interoperability which in cooperation with the University of Kentucky and Amoco Production Company Research Center in November, 1990.

3 Private communication with Lotus Development Corporation's Datalens industry marketing team, October 1990.

the semantics of data provided by databases are meaningful to the application. Methods were proposed for detecting changes in data semantics and for determining if the databases can continue to supply meaningful data to the application. In these research efforts, the bottom-up notion is imbedded and the word "mod[illegible]" is used in the context of data modeling.

Most other research efforts in the area of heterogeneous database systems have focused on system building, transaction management, and query processing (Breitbart, Olson, & Thompson, 1986; Brill, Templeton, & Yu, 1984; Czejdo, Rusinkiewicz, & Embley, 1987; Deen, Amin, & C., 1987; Elmagarmid, et al., 1990; Ferrier & Strangret, 1982; Heimbigner & McLeod, 1985; Litwin & Abdellatif, 1986; Litwin, et al., 1982; Lyngback & McLeod, 1983; Smith, et al., 1981; Wang & Madnick, 1988). A common assumption of all these research efforts is that semantic heterogeneity will be reconciled through schema integration. However, work on schema integration (Batini, Lenzirini, & Navathe, 1986; Dayal & Hwang, 1984; Elmasri, Larson, & Navathe, 1987) has also been bottom-up oriented. A group of Data Base Administrators (DBAs) would get together to compare the entities, relationships, and attributes in their local databases. After figuring out the differences in their local databases, an integrated schema would be proposed, together with the mapped relationships between the integrated schema and the corresponding local schemas. Although appropriate for integrating a small number of local databases, the task becomes formidable as the number of local databases increases (Batini, Lenzirini, & Navathe, 1986; Elmasri, Larson, & Navathe, 1987). More important, the DBAs may integrate local databases from only the technical viewpoint without fully considering managerial implications.

In contrast, a top-down, application model-based approach would enable us to focus on the data required by this application model, thus reducing the scope of integration. Application models will also meet the needs of application users and provide a more natural and stable data environment.

RESEARCH GOAL AND CONTRIBUTION

The goal of this research is to develop a model-based approach for reconciling semantic heterogeneity of data in heterogeneous database systems. In this paper, we present an accounting

model-based approach for producing integrated financial statements from databases which are being used by practitioners. The accounting model is built from general accounting knowledge (Foster, 1986; Kohler, 1983). For illustrative purpose, we focus on accounting knowledge pertinent for constructing financial statements.

The significance of this paper is as follows: A top-down, accounting model-based approach is presented to reconcile semantic heterogeneity among the data retrieved from multiple financial accounting databases. Knowledge represented in this model is applied to reconcile data conflicts and to infer new information. Accounting model-based rules for reconciling semantic heterogeneity at the data level are illustrated with intriguing cases using real data.

The concept of knowledge representation for model management systems has been examined in the context of mathematical programming and other areas (Bhargava & Kimbrough, 1990; Bhargava, Kimbrough, & Krishnan, 1989; Dolk, 1986; Dolk, 1988; Dolk & Konsynski, 1984). However, to the best of our knowledge, this approach has not been applied to the heterogeneous database systems area for reconciling semantic heterogeneity at the data level as well as the schema level.

By virtue of its rigorous nature, the accounting discipline lends itself to a solid foundation for reconciling semantic heterogeneity inherent in disparate financial accounting databases. The requirement to balance all the accounts has provided a model-based approach for evaluating the reliability and validity of data retrieved from heterogeneous database systems. The top-down, accounting model-based approach has further enabled financial analysts to focus on the application requirements that most concerned them: utilizing integrated financial statements for tasks such as profitability analysis.

RESEARCH BACKGROUND AND ASSUMPTIONS

We have evolved a heterogeneous database system with source tagging capabilities (Codes, 1989; Gupta, et al., 1989; Madnick, Siegel, & Wang, 1990; Paget, 1989; Wang & Madnick, 1988; Wang & Madnick, 1990; Yuan, 1990). The query processor architecture is depicted in Figure 1. Briefly, the Application Query Processor (AQP) translates an end-user query into a polygen query for the Polygen

Query Processor (PQP) based on the user's application schema. The term "polygen" is used (instead of the more conventional term - "global") to signify a query processor with source tagging capabilities. The PQP in turn translates the polygen query into a set of local queries based on the corresponding polygen schema, and routes them to the Local Query Processors (LQP). The details of the mapping and communication mechanisms between an LQP and its local database is encapsulated in the LQP. Upon return from the LQPs, the retrieved data are further processed by the PQP in order to produce the desired composite information.

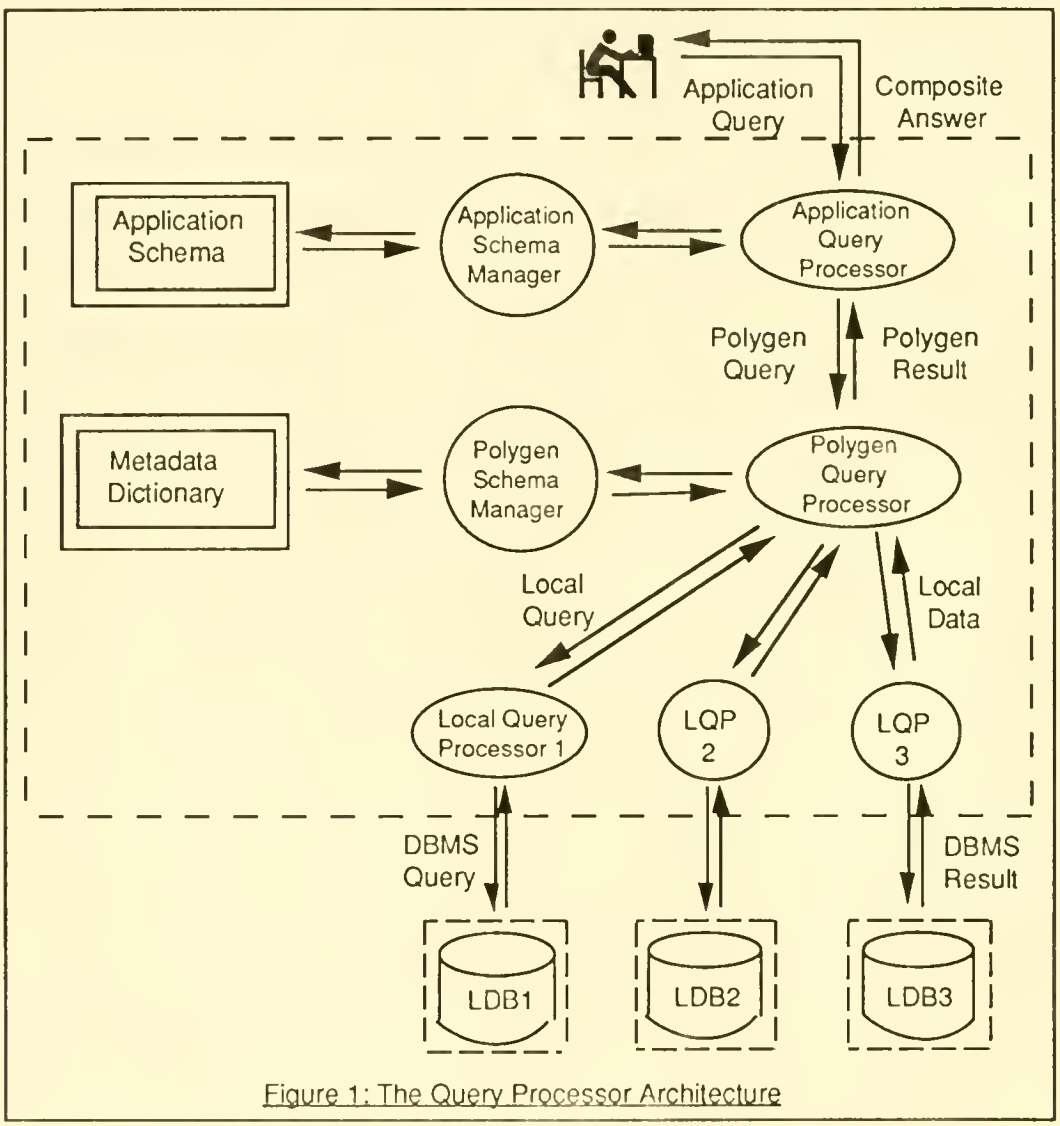

Figure 1: The Query Processor Architecture

Section II presents an accounting model-based approach. Section III addresses issues involved in reconciling semantic heterogeneity at the schema level based on the knowledge represented in the accounting model. Section IV illustrates how semantic heterogeneity at the account-data level can be reconciled. In Section V, a profitability analysis is presented based on financial ratios. Finally, concluding remarks are made in Section VI.

II. An Accounting Model-Based Approach

Financial statements such as the balance sheet, income statement, and cash flow statement are the most widely distributed accounting information. Financial analysts routinely utilize such information as a starting point to analyze the financial status of companies. Since accounting terminology, such as accounts receivable, are well understood by accountants, an accounting model could be constructed for a heterogeneous database system.

An Entity-Relationship (ER) view of accounting models (McCarthy, 1979; McCarthy, 1982) was proposed as a generalized approach for both accountants and non-accountants. Correspondence of the approach with the accounting theories was discussed. Applications of this approach in the database design phases of view modeling and view integration were presented. Finally, ER representations of accounting objects were reconciled with those representations found in conventional double-entry systems. This approach deals with problems in a single database environment and at the schema level. Efforts to extend this ER view of accounting models for reconciling data values retrieved from various data sources can be found elsewhere (Chen & Wang, 1990). In this paper, we focus on the accounting aspect.

Briefly, total assets, total liabilities, and shareholders' equity are three primary accounts in the balance sheet. For example, total assets equals total current assets plus total non-current assets. Each account, in turn, is computed from other more detailed accounts, as shown in Figure 2. The hierarchical structure exhibits the inter-relationship of accounts. We call the highest level account in each hierarchical structure a "root account." An account is called a "parent" account in relation to its immediate lower level accounts; each of the lower level accounts is called a "child" account. A parent

account is more aggregated than its children accounts. Thus, on the one hand, we attain more detailed information as we move down a hierarchy; on the other hand, we attain more aggregated information as we move up a hierarchy.

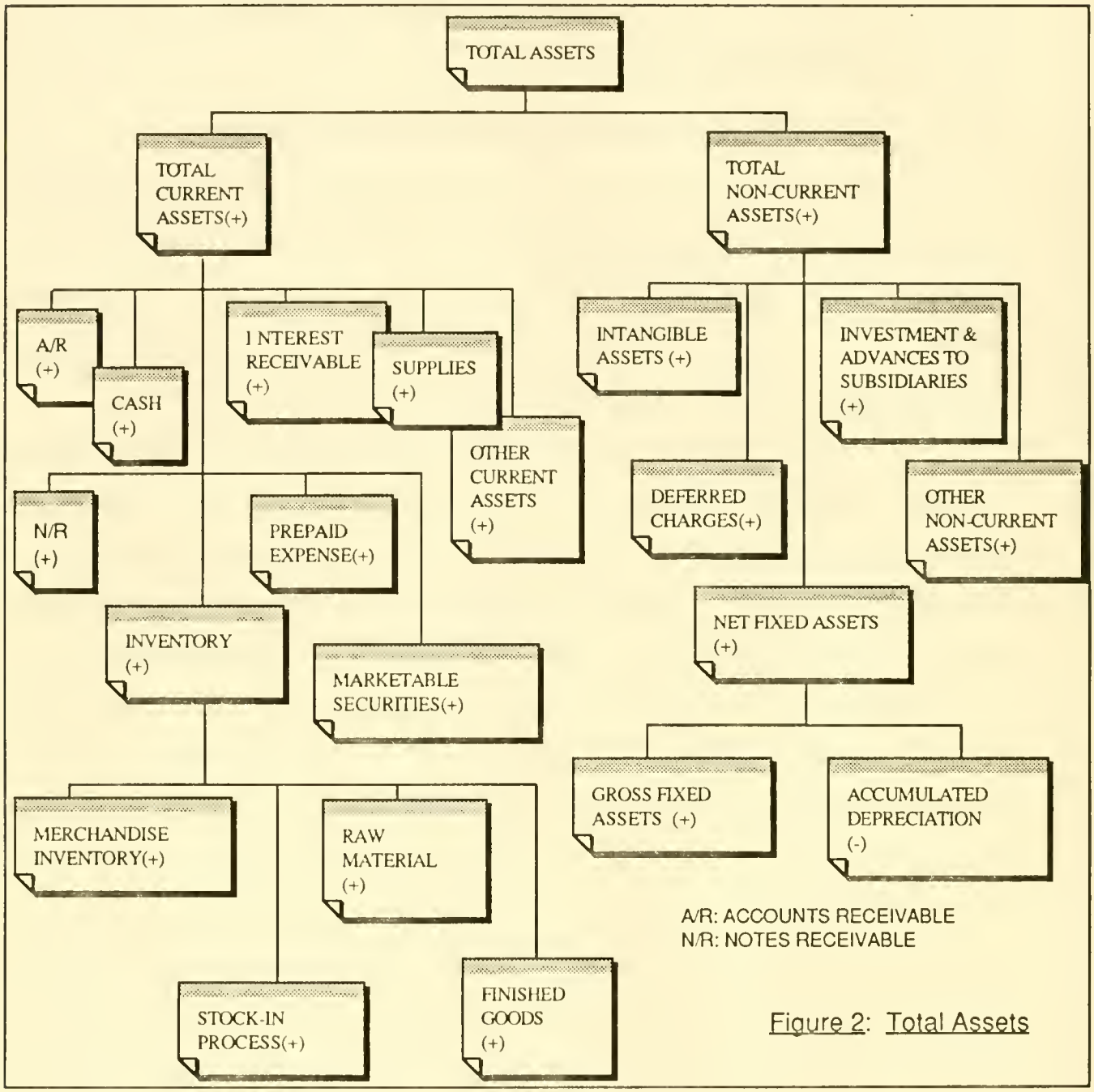

Figure 2: Total Assets

RELATIONSHIP REPRESENTATION

Broadly, the inter-relationships of the various accounts can be classified into two categories: explicit and implicit.

Explicit Relationship. The relationship between the parent account and the child account is represented by an arithmetic sign in parentheses, such as (+) or (-). For example, in Figure 2, the explicit relationship of net fixed assets, gross fixed assets, and accumulated depreciation is "Net Fixed Assets" = "Gross Fixed Assets" - "Accumulated Depreciation."

The income statement and the balance sheet are linked together because net income, the root account in the income statement, is a child account of retained earnings in the balance sheet.[4] The cash flow statement is derived from the balance sheet and the income statement.

Implicit Relationship. The hierarchical inheritance property can be exploited. From Figure 3 an application system can determine that:

cash flow = cash flow from operation + cash flow from finance + disposal of fixed assets - acquisition of fixed assets - other capital expenditure

As another example, in addition to the relationship of "Net Fixed Assets" = "Gross Fixed Assets" - "Accumulated Depreciation" mentioned earlier, net fixed assets should be less than gross fixed Assets. This relationship is based on the supposition that as long as a firm has fixed assets, its gross fixed assets and accumulated depreciation will have positive value and its accumulated depreciation will not exceed gross fixed assets.

Figures 3 and 4 summarize the relationships of the financial statements.

---

4 This comes from the relationship of "Retained Earnings" = "Retained Earnings at Beginning" + "Net Income" - "Dividend".

Income Statement	Cash flow Statement
N1. Net Sales	C1. Net Income (N16)
N2. Cost of Goods Sold	C2. Deferred Tax (L10)
N3. Gross Profit (N1-N2)	C3. Depreciation Expense (N7)
N4. R&D Expense	C4. Change in Working Capital
N5. Selling & Administration	C5. Other Cash flow From Operation
N6. Operating Income (N3-N4-N5)	C6. Cash Flow From Operation (C1+C2+C3+(-)C4+C5)
N7. Depreciation	C7. Acquisition of Fixed Assets
N8. Non-Operating Income	C8. Disposal of Fixed Assets
N9. Interest Expense	C9. Other Capital Expenditure
N10. Net Income Before Tax (N6-N7+N8-N9)	C10. Cash Flow From Investment (-C7+C8-C9)
N11. Tax Expense	C11. Net Increase in Short-Term Debt
N12. Minority Interest	C12. Insurance of Common Stock
N13. Other Income	C13. Increase of Treasury Stock
N14. Net Income Before Extraordinary Income (N10-N11-N12+N13)	C14. Repayment of Long-Term Debt
	C15. Issuance of Long-Term Debt
N15. Extraordinary Income	C16. Cash Flow From Finance (C11+C12-C13-C14+C15)
N16. Net Income (N14+N15)	C17. Total Cash Flow (C6+C10+C16)

Figure 3 Income Statement & Cash flow Statement

A1. Cash	L1. Accounts Payable
A2. Marketable Securities	L2. Note Payable
A3. Accounts Receivable	L3. Tax Payable
A4. Notes receivable	L4. Interest Payable
A5. Inventory (A5.1+A5.2+A5.3+A5.4)	L5. Accrued Expenses
A5.1 Merchandise Inventory	L6. Advances
A5.2 Stock In Progress	L7. Other Current Liabilities
A5.3 Raw Material	L8. Total Current Liabilities (L1+L2+L3+L4+L5+L6+L7)
A5.4 Finished Goods	L9. Long-Term Debt (L9.1+L9.2+L9.3)
A6. Interest Receivable	L9.1 Mortgage Payable
A7. Supplies	L9.2 Bond Payable
A8. Prepaid Expense	L9.3 Capitalized Lease
A9. Other Current Assets	L10. Deferred Tax
A10. Total Current Assets	L11. Deferred Liabilities
A10. Total Current Assets (A1+A2+A3+A4+A5+A6+A7+A8+A9)	L12. Other Non-Current Liabilities
	L13. Total Non-Current Liabilities (L9+L10+L11+L12)
A11. Gross Fixed Assets	L14. Total Liabilities (L8+L13)
A12. Accumulated Depreciation	M1. Minority Interest
A13. Net Fixed Assets (A11-A12)	E1. Preferred Stock
A14. Investment & Advances to subsidiaries	E2. Common Stock
A15. Deferred Charges	E3. Additional Paid-in Capital
A16. Intangible Assets	E4. Treasury Stock
A17. Other Non-Current Assets	E5. Retained Earnings (E5.1+E5.2-E5.3)
A18. Total Non-Current Assets (A13+A14+A15+A16+A17)	E5.1 Retained Earning at Beginning
	E5.2 Net Income (N16)
A19. Total Assets (A10+A18)	E5.3 Dividend
A19=L14+M1+E6	E6. Total Shareholders' Equity (E1+E2+E3+E4+E5)

Figure 4 Balance Sheet

Two separate tasks, schema integration and query processing, need to be performed in applying a top-down, accounting model-based approach to producing integrated financial statements from multiple databases. During schema integration, the semantic heterogeneity of entities, attributes, and relationships are reconciled. During query processing, the semantic heterogeneity of data values are reconciled. We focus on the schema-level heterogeneity in Section III, and data-level heterogeneity in Section IV.

III. Reconciling Schema-Level Semantic Heterogeneity

We first introduce three financial accounting databases which will be used to present an application example in this paper. They are Finsbury's Dataline, I.P. Sharp's Disclosure, and Lotus' LotusOne. As mentioned earlier, these databases are currently being used by practitioners to perform financial analysis.

Dataline provides various financial statements from the previous five years for more than three thousand companies in Europe and Japan. The financial statements include balance sheets, income statements, and major accounting ratios.

Disclosure contains financial and management information from more than 12,000 public companies which file reports with the U.S. Securities and Exchange Commission (SEC). As such, companies whose information is provided in Disclosure have at least 500 shareholders of one class of stock and at least $5 million in assets. The financial data provided by Disclosure includes quarterly and yearly financial statements such as balance sheets, income statements, changes in financial status, and key financial ratios.

LotusOne also provides financial and management information about companies which report to the SEC in the U.S. Since the information in LotusOne is quite similar to that of Disclosure, it is easy to illustrate important semantic heterogeneity by comparing the overlapping information.

As an example, we focus on the financial statements of Volvo Corporation based on raw data from the three databases.

SCHEMA INTEGRATION

Many issues need to be addressed in schema integration of multiple financial accounting databases either with or without a top-down, accounting model-based approach. The difference is that an accounting model-based approach enables us to reconcile semantic heterogeneity among disparate databases based on the accounts represented in the model and their relationships. We have found that the arrangement and choice of accounting terms of these financial accounting databases are so incompatible that even an experienced accountant would take a fair amount of time in order to reconcile the semantic heterogeneity. Table 1 shows the reconciled total assets for these three databases.

Table 1: Metadata for Total Assets

Federated Account	LotusOne	Disclosure	Dataline
A1. Cash	Cash	Cash & Equivalents (?)	Cash & Equivalents
A2. Marketable Securities	Marketable Securities		Marketable Securities
A3. Accounts Receivable	Accounts Receivable	Accounts Receivable	Debtors (?)
A4. Notes receivable	Notes receivable	Notes receivable	
A5. Inventory	Inventory	Inventory	Inventory
A5.1 Merchandise Inventory	NA	NA	NA
A5.2 Stock In Progress	Stock In Progress	NA	Stock In Progress
A5.3 Raw Material	Raw Material	NA	Raw Material
A5.4 Finished Goods	Finished Goods	NA	Finished Goods
A6. Interest Receivable	Other Current Assets & Prepaid Expense (?)	Other Current Assets (?)	NA
A7. Supplies			
A8. Prepaid Expense			
A9. Other Current Assets			
A10. Total Current Assets	Total Current Assets	Total Current Assets	Total Current Assets
A11. Gross Fixed Assets	Gross Fixed Assets	Gross Fixed Assets	Gross Fixed Assets
A12. Accumulated Depreciation	Accumulated Depreciation	Accumulated Depreciation	Accumulated Depreciation
A13. Net Fixed Assets	Net Fixed Assets	Net Fixed Assets	Net Fixed Assets
A14. Investment & Advances to subsidiaries	Investment & Advances to subsidiaries	Investment & Advances to subsidiaries	Investment & Advances to subsidiaries
A15. Deferred Charges	Deferred Charges	Deferred Charges	NA
A16. Intangible Assets	Intangibles	Intangible Assets	NA
A17. Other Non-Current Assets	Other Non-Current Assets	Other Non-Current Assets	NA
A18. Total Non-Current Assets	Total Non-Current Assets	Total Non-Current Assets	Total Non-Current Assets
A19. Total Assets	Total Assets	Total Assets	Total Assets

In Table 1, if the database provides data for the parent account, but not for the child account due to different levels of aggregation, then "NA" (not applicable) is recorded. If an account does not

match any accounts employed in the model, but the parent account can be identified, then a question mark is entered. For example, in the case of "Debtors" from Dataline, we recognize that this account is the aggregation of "Accounts Receivable" and "Notes receivable." Thus, it belongs to the parent account "Current Assets." However, we cannot specify the exact matching level of the account in the accounting model. Therefore, this account is marked with "?." Accounts in the model which do not have any matching accounts from any databases are filled with "NA." This phenomenon will happen because the model has a more detailed level of aggregation than the actual databases under investigation.

Some typical schema integration problems (Batini, Lenzirini, & Navathe, 1986) in the accounting domain are exemplified below.

Same Terms for Different Concepts. In Dataline, the term "Cash + Equivalents" is used to represent "Cash." In Disclosure, the term "Cash & Equivalents" is used to include the "Cash" and "Marketable Securities."

Different Accounting Terms. Different accounting terms are used to represent the same concept. For example, "NI Before Extraordinary Income" is used in LotusOne, "Net Income Before Ext" in Disclosure, and "Earned for Ordinary" in Dataline. As another example, "Shareholders' Equity" is used in LotusOne, "Total Equity" in Disclosure, and "Capital and Reserves" in Dataline.

Account Incompleteness. LotusOne and Disclosure contain accounts such R&D expenses and Selling & Administration Expenses, whereas Dataline does not.[5]

Different Levels of Aggregation. LotusOne and Disclosure provide aggregated data for total gross fixed assets, whereas Dataline provides data for each item of gross fixed assets, such as land, buildings, plants and machinery.

These issues are reconciled during the schema integration process following the accounting model. The knowledge of schema level semantic heterogeneity is recorded in a metadata dictionary. For example, Table 1 forms part of the metadata dictionary.

---

5 There are also cases in which although the database may have a specific account, it has no data for it. For instance, although all three databases have an item for "Accumulated Depreciation," LotusOne and Disclosure do not have data in for it. This account-value level issue will be addressed at run-time by the query processor.

During run-time, the metadata dictionary is used by the query processor to match the accounts from the accounting terms used in each database so that data can be retrieved from these databases. After the data have been retrieved, reliability and credibility of the data need to be examined before integrated financial statements can be produced.

The following section presents the solution techniques that we have developed.

IV. Reconciling Data-Level Semantic Heterogeneity

At run time, a copy of the accounting model is created to hold the data retrieved from each of the financial databases. As long as the data for the aggregated accounts are semantically consistent, we will ignore the semantic conflicts which may occur in the lower level accounts. In doing so, we have assumed that the data for the children accounts will not be needed later for further financial analysis. In general, depending on the users' requirements, we may have to obtain data for a child account. For example, Dataline does not provide data for "Accounts Receivable" but does provide the sum of "Accounts Receivable" and "Notes Receivable" under the name of "Debtors." If we have to calculate the accounts-receivable turn over ratio, then a less aggregated child account will be needed. Although we still won't be able to obtain data for accounts receivable from Dataline in that case, the data for debtors can be used to verify the data from LotusOne and Disclosure.

Tables 2 and 3 present the data for the balance sheet and the income statement of Volvo from LotusOne, Disclosure, and Dataline respectively. At this stage, the data for each account will be compared and the three financial statements will be merged into one.

Table 2: Comparison of the Balance Sheet for Volvo (1988)

Balance Sheet	LotusOne	Disclosure	Dataline
Total Current Assets	48978000	48.98	48978
Gross Fixed Assets	15610000	15.61	30236
Accumulated Depreciation	NA	NA	14626
Net Fixed Assets	15610000	15.61	15610
Investment & Advances to subsidiaries	14068000	14.07	14068
Deferred Charges	NA	NA	NA
Intangible Assets	297000	0.3	297
Other Non-Current Assets	7998000	8	7998
Total Non-Current Assets	NA	NA	NA
Total Assets	86951000	86.95	86951
Liability & Shareholder's Equity	LotusOne	Disclosure	Dataline
Total Current Liabilities	34500000	34.5	34500
Long-Term Debt	6758000	6.76	6758
Deferred Liabilities	NA	NA	30375
Other Non-Current Liabilities	30375000	3038	NA
Total Non-Current Liabilities	NA	NA	NA
Total Liabilities	71633000	71.63	71633
Minority Interest	484000	0.48	484
Shareholder's Equity	14834000	14.83	14834
Total	86951000	86.95	86951

Table 3: Comparison of the Income Statement for Volvo (1988)

Income Statement	LotusOne	Disclosure	Dataline
Net Sales	96639000	96.64	96639
Cost of Goods Sold	77110000	77.11	NA
Gross Profit	19529000	19.53	7186
R&D Expenses	NA	NA	NA
Selling & Administration	10028000	10.03	NA
Operating Income	9501000	9.5	NA
Depreciation	2293000	2.29	2293
Non-Operating Income	685000	0.69	NA
Interest Expense	1961000	1.96	1961
Net Income Before Tax	5932000	5.93	5932
Tax Expense	2500000	2.5	2500
Minority Interest	103000	0.1	103
Other Income	NA	NA	NA
Net Income Before Extraordinary Items	3329000	3.33	3329
Extraordinary Items	NA	NA	NA
Net Income	33290009	NA	3329

UNIT CONVERSION

The three databases use different units in representing values for each account. Consequently, the values for the same account appear to be different. For example, intangible assets is recorded as 297000 in LotusOne, 0.3 in Disclosure, and 297 in Dataline. These three values appear to be inconsistent. However, they are essentially the same because LotusOne reports the data in Swedish Kronor, Dataline in thousands of Swedish kronor, and Disclosure in millions of Swedish kronor (round up). Therefore, before moving on to the next step, the units should be adjusted so that the account values from each database can be compared. Research issues related to unit conversion has been addressed elsewhere (Rigaldies, 1990).

MODEL-BASED JUDGEMENT

The conflicts at the account data level can be classified into two categories: (1) databases provide inconsistent data for the same entities, and (2) account data for certain databases are not available.

In reconciling these conflicts, we need certain rules to evaluate the reliability of the data from each database. We apply the accounting model presented in Section II to reconcile conflicts at the account data level. The two examples of data level conflicts below illustrate how the explicit and implicit relationships represented in the accounting model can be exploited.

Example 1. As shown in Table 2, Lotus reports 15610000 for gross fixed assets, Disclosure 15.61, and Dataline 30236.

At first glance, it appears that the difference between 15610000 and 15.61 is due to unit difference and Dataline's 30236 is a typo. Therefore, one may conclude that 15.61 millions should be used for gross fixed assets.

However, from the accounting model, we can find that "Net Fixed Assets" = "Gross Fixed Assets" - "Accumulated Depreciation." From this relationship, we can infer which data is more reliable.

Both LotusOne and Disclosure do not provide data for "Accumulated Depreciation" and contain

the same number for both "Gross Fixed Assets" and "Net Fixed Assets" which does not satisfy the implicit relationship that gross fixed assets should be less than net fixed assets. In contrast, Dataline not only provides data for all of the three accounts but also provides data values which satisfy the relationship. From this fact, one would infer that Dataline provides more reliable data for "Gross Fixed Assets" and "Accumulated Depreciation." Therefore, 30236 thousands should be used instead.

Example 2. As shown in Table 3, Lotus reports 19529000 for gross profit, Disclosure 19.53, and Dataline 7186. The data for "Gross profit" from Dataline are less reliable than those of LotusOne and Disclosure for the following reasons:

- The items which are associated with "Gross Profit" are "Cost of Goods Sold" and "Net Sales" (See Figure 3). Since Dataline does not provide any data for "Cost of Goods Sold" which is required to calculate "Gross Profit," we cannot verify the figure for "Gross Profit" from Dataline.
- The other two databases, LotusOne and Disclosure, provide data for both "Cost of Goods Sold" and "Net Sales" and the data for "Gross Profit" is consistent with the explicit relationship of "Gross Profit" = "Net Sales" - "Cost Of Goods Sold."

The above observation suggests that none of the databases dominate others in terms of reliability. Consequently, when we encounter account data level inconsistencies, we should infer data reliability, case by case, with the help of the following rules.

1. Use the knowledge represented in the accounting model to find the relationship which links the associated accounts to the account in question.
2. Verify the data from each database by using the relationship. In doing so, pay attention to the implicit relationships as well as the explicit relationships.
3. If the values from certain databases do not satisfy the relationship, then discard them.
4. If no associated item or relationship can be found, or the inconsistencies cannot be reconciled through the above-mentioned steps, then apply the majority rule.

These rules have been successfully applied to all of the data inconsistencies at the account value level in the process of producing integrated financial statements. The integrated balance sheet and income statement for Volvo is presented in Table 4 and Table 5 respectively.

Table 4: Integrated Balance Sheet for Volvo (1988)

Balance Sheet	Composite Data (Swedish Kronor in thousands)
Total Current Assets	48978
Gross Fixed Assets	30236
Accumulated Depreciation	14626
Net Fixed Assets	15610
Investment & Advances to subsidiaries	14068
Deferred Charges	NA
Intangible Assets	297
Other Non-Current Assets	7998
Total Non-Current Assets	37973
Total Assets	86951
Liability & Shareholder's Equity	
Total Current Liabilities	34500
Long-Term Debt	6758
Deferred Liabilities	NA
Other Non-Current Liabilities	30375
Total Non-Current Liabilities	NA
Total Liabilities	71633
Minority Interest	484
Shareholder's Equity	14834
Total	86951

Table 5: Integrated Income Statement for Volvo (1988)

Income Statement	Composite Data (Swedish Kronor in thousands)
Net Sales	96639
Cost of Goods Sold	77110
Gross Profit	19529
R&D Expenses	NA
Selling & Administration	10028
Operating Income	9501
Depreciation	2293
Non-Operating Income	685
Interest Expense	1961
Net Income Before Tax	5932
Tax Expense	2500
Minority Interest	103
Other Income	NA
Net Income Before Extraordinary Items	3329
Extraordinary Items	NA
Net Income	33290

Following the same process used to obtain the integrated statements for Volvo, we have constructed the integrated financial statements for both Ford and Honda, as shown in Tables 6-7 and

Tables 8-9 respectively. We are now in a position to compute profitability ratios for each of these three companies, as discussed in the following section.

Table 6: Integrated Balance Sheet for Ford (1988)

Balance Sheet	Composite Data (US dollars in thousands)
Total Current Assets	118888600
Gross Fixed Assets	15992200
Accumulated Depreciation	NA
Net Fixed Assets	15992200
Investment & Advances to subsidiaries	2102700
Intangible Assets	NA
Other Non-Current Assets	6383000
Total Non-Current Assets	24477900
Total Assets	143366500
Liability & Shareholder's Equity	
Total Current Liabilities	40959700
Long-Term Debt	68187400
Deferred Liabilities	3933100
Other Non-Current Liabilities	8593200
Total Non-Current Liabilities	80713700
Total Liabilities	121673400
Minority Interest	NA
Shareholder's Equity	21529000
Total	143366500

Table 7: Integrated Income Statement for Ford (1988)

Income Statement	Composite Data (US dollars in thousands)
Net Sales	92445600
Cost of Goods Sold	74017300
Gross Profit	18428300
R&D Expenses	NA
Selling & Administration	6972500
Operating Income	11455800
Depreciation	3792300
Non-Operating Income	1033000
Interest Expense	354000
Net Income Before Tax	8342500
Tax Expense	2998700
Minority Interest	43600
Other Income	NA
Net Income Before Extraordinary Items	53000200
Extraordinary Items	NA
Net Income	53000200

Table 8: Integrated Balance Sheet for Honda (1988)

Balance Sheet	Composite Data (Japanese Yens in thousands)
Total Current Assets	1269160
Gross Fixed Assets	1637345
Accumulated Depreciation	771250
Net Fixed Assets	866095
Investment & Advances to subsidiaries	117447
Intangible Assets	0
Other Non-Current Assets	31747
Total Non-Current Assets	1014839
Total Assets	2284449
Liability & Shareholder's Equity	
Total Current Liabilities	1047370
Long-Term Debt	301572
Deferred Liabilities	34049
Other Non-Current Liabilities	0
Total Non-Current Liabilities	945621
Total Liabilities	1382991
Minority Interest	NA
Shareholder's Equity	901458
Total	2284449

Table 9: Integrated Income Statement for Honda (1988)

Income Statement	Composite Data (Japanese Yens in thousands)
Net Sales	3489258
Cost of Goods Sold	2544188
Gross Profit	945070
R&D Expenses	183652
Selling & Administration	584360
Operating Income	177058
Depreciation	NA
Non-Operating Income	19578
Interest Expense	24547
Net Income Before Tax	172089
Tax Expense	80559
Minority Interest	NA
Other Income	5769
Net Income Before Extraordinary Items	97299
Extraordinary Items	NA
Net Income	97299

V. Profitability Analysis

The most widely-used profitability ratios are Gross Profit Rate (GPR), ROE (Return On Equity), ROA (Return On Asset), and ROS (Return On Sales). Table 10 shows these popular profitability ratios.

Table 10: Profitability Ratios

Ratio	Definition	Financial Statement
GPR	Gross Profit/Net Sales	Income Statement
ROE	Income Before Tax /Shareholder's Equity	Income Statement & Balance Sheet
ROA	Income Before Tax/Total Assets	Income Statement & Balance Sheet
ROS	Income Before Tax/Net Sales	Income Statement

Based on the definition shown in Table 10 and the integrated financial statements for the three companies shown in Tables 4-9, we can compute the profitability ratios for Ford, Honda, and Volvo. The results are shown in Table 11. Note that by using ratios, we do not have to deal with the issue that the currency used in the Volvo financial statements is kronor in thousands, Ford is dollars in thousands, and Honda is yens in thousands. However, in other financial analyses, such as a direct comparison of gross profit, one would need the exchange rates of these currencies.

Table 11: Profitability Ratios for Ford, Honda, and Volvo (1988)

Ratios	Ford	Honda	Volvo
GPR	0.199	0.271	0.202
ROE	0.388	0.191	0.4
ROA	0.058	0.075	0.068
ROS	0.09	0.049	0.061

According to the figures for ROE and ROS, Ford can be said to be very profitable. However, of the three ratios, the GPR is the lowest for Ford. Because "Gross Profit" is calculated by subtracting the "Cost Of Goods Sold" from the "Net Sales", it can be said that the profit margin of Ford is low due to the high cost of goods sold. From the above argument, it could be concluded that due to high non-operating income, Ford is profitable, but due to the high cost of goods sold, it has a low GPR.

The above analysis is based on the assumption that all three companies have used the same accounting principles. That is, the same inventory evaluation method and depreciation method is used in the financial statements of all three companies. In reality, they use different accounting principles.

Since the cost of goods sold and income before tax depends on the accounting principles adopted in the inventory and depreciation evaluation, the financial ratios are functions of the accounting principles. The discussion of these issues is beyond the scope of this paper.

VI. Concluding Remarks

We have presented a top-down, accounting model-based approach for producing integrated balance sheets and income statements. In addition, profitability ratios for Volvo, Ford, and Honda have been computed based on the integrated financial statements composed of data from Finsbury's Dataline, I.P. Sharp's Disclosure, and Lotus' LotusOne databases. This approach has enabled financial analysts to focus on the application requirements that most concerned them, namely utilizing integrated financial statements for tasks such as profitability analysis.

We have focused on the fundamental accounting principles pertinent to financial statement construction. By virtue of its rigorous nature, the accounting discipline lends itself to a solid foundation for reconciling semantic heterogeneity inherent in disparate financial accounting databases. The requirement to balance all the accounts has enabled us to evaluate the reliability and validity of financial data retrieved from heterogeneous database systems.

Our research thrust is to provide a model-based interface for business applications. The model-based approach checks the reliability and validity of data retrieved from various data sources using knowledge encoded in financial models. This would facilitate access to and seamless integration of financial data stored in heterogeneous database systems. We believe that this effort will not only contribute to the academic research frontier but also benefit the business community in the foreseeable future.

VI. Bibliography

[1] Batini, C., Lenzirini, M., & Navathe, S. (1986). A comparative analysis of methodologies for database schema integration. *ACM Computing Survey, 18*(4), pp. 323 - 364.

[2] Bhargava, H. & Kimbrough, S. (1990). *On Embedded Languages for Model Management.* Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences. Hawaii. January 1990.

[3] Bhargava, H., Kimbrough, S., & Krishnan, R. (1989). Unique Names Violations: A Problem for Model Integration or You Say Tomato, I Say Tomahto. *To Appear in ORSA, Journal on Computing,.*

[4] Breitbart, Y., Olson, P. L., & Thompson, G. R. (1986). *Database integration in a distributed heterogeneous database system.* Los Angeles, CA. February 1986. pp. 301-310.

[5] Brill, D., Templeton, M., & Yu, C. (1984). *Distributed query processing strategies in MERMAID, a frontend to data management systems.* First International Conference on Data Engineering. Los Angeles, CA. February 1984. pp. 301-310.

[6] Chen, P. & Wang, Y. R. (1990). *An Entity-Relationship Approach for Accounting Systems with Multiple Data Sources.* (CIS-90-32) MIT Sloan School of Management. October 1990.

[7] Czejdo, B., Rusinkiewicz, M., & Embley, D. (1987). *An approach to schema integration and query formulation in federated database systems.* The 3rd International Conference on Data Engineering. Los Angeles, CA. 1987. pp. 477-484.

[8] Dayal, U. & Hwang, K. (1984). View definition and generalization for database integration in multidatabase system. *IEEE Transactions on Software Engineering, SE-10,* pp. 628-644.

[9] Deen, S. M., Amin, R. R., & C., T. M. (1987). Data integration in distributed databases. *IEEE Transactions on Software Engineering, SE-13,* pp. 860-864.

[10] Dolk, D. R. (1986). A Generalized Model Management System for Mathematical Programming. *ACM Transactions on Mathematical Software, 12*(2), pp. 92-126.

[11] Dolk, D. R. (1988). Model Management and Structured Modeling: The Role of an Information Resource Dictionary System. *Communications of the ACM, 31*(6), pp. 704-718.

[12] Dolk, D. R. & Konsynski, B. R. (1984). Knowledge Representation for Model Management System. *IEEE Transactions on Software Engineering, SE-10*(6), pp. 619-628.

[13] Elmagarmid, A. K., et al. (1990). *A multidatabase transaction model for InterBase.* To Appear in the 16th International Conference on Very Large Data Bases. Brisbane, Australia. August 1990.

[14] Elmasri, R., Larson, J., & Navathe, S. (1987). *Schema integration algorithms for federated databases and logical database design.* Honeywell Inc., Submitted for Publication. 1987.

[15] Ferrier, A. & Strangret, C. (1982). *Heterogenity in the distributed database management systems Sirius-Delta.* The 8th International Conference on Very Large Data Bases. Mexico City, Mexico. 1982.

[16] Foster, G. (1986). *Financial Statement Analysis* (2nd ed.). New Jersey: Prentice - Hall, Inc.

[17] Godes, D. B. (1989). *Use of heterogeneous data sources: three case studies.* (CIS-89-02) Sloan School of Management, MIT, Cambridge, MA. June 1989.

[18] Goodhue, D. L., Quillard, J. A., & Rockart, J. F. (1988). Managing The Data Resources: A Contingency Perspective. *MIS Quarterly, 12*(3), pp. 373-392.

[19] Gupta, A., et al. (1989). *An architecture comparison of contemporary approaches and products for*

integrating heterogeneous information systems. Sloan School of Management, MIT, Cambridge, MA 02139. 1989.

[20] Heimbigner, D. & McLeod, D. (1985). A Federated architecture for information management. *ACM Transactions on Office Information Systems, 3*, pp. 253-278.

[21] Kohler, E. L. (1983). *A Dictionary for Accountants* . New Jersey: Prentice - Hall, Inc.

[22] Litwin, W. & Abdellatif, A. (1986). Multidatabase interoperability. *IEEE Computer, ,* pp. 10-18.

[23] Litwin, W., et al. (1982). *SIRIUS system for distributed data management.* International Symposium on Distributed Databases. Berlin, West Germany. 1982. pp. 311-366.

[24] Lyngbaek, P. & McLeod, D. (1983). *An approach to object sharing in distributed database systems.* The 9th International Conference on Very Large Data Bases. October 1983. pp. 364-374.

[25] Madnick, S. E., Siegel, M., & Wang, Y. R. (1990). The Composite Information Systems Laboratory (CISL) project at MIT. *IEEE Data Engineering, 13*(2), pp. 10-15.

[26] Madnick, S E. & Wang, Y. R. (1986). *Key Concerns of Executives Making IS Decisions.* 21st Hawaii International Conference on System Sciences. Kailu-Kona, Hawaii. January 1986. pp. 254-260.

[27] McCarthy, W. E. (1979). An Entity-Relationship View of Accounting Models. *The Accounting Review, 54*(4), pp. 667-686.

[28] McCarthy, W. E. (1982). The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. *The Accounting Review, 57*(3), pp. 667-686.

[29] Paget, M. L. (1989). *A knowledge-based approach toward integrating international on-line databases.* (CIS-89-01) Sloan School of Management, MIT, Cambridge, MA. May 1989.

[30] Reiner, D. (1990). Special Issue on Database Connectivity. *IEEE Data Engineering, 13*(2), pp. 52.

[31] Rigaldies, B. (1990). *Technologies and Policies for the Development of CIS in Decentralized Organizations.* (WP# CIS-90-05) Master Thesis, MIT Sloan School of Management. May 1990.

[32] Rockart, J. F. & Short, J. E. (1989). IT in the 1990s: Managing Organizational Interdependence. *Sloan Management Review, Sloan School of Management, MIT, 30*(2), pp. 7-17.

[33] Smith, J. M., et al. (1981). *Multibase - Integrating Heterogeneous Distributed Database Systems.* 1981 National Computer Conference. 1981. pp. 487-499.

[34] Wang, Y. R. & Madnick, S. E. (1988). *Connectivity among information systems* . Cambridge, Mass.: Composite Information Systems (CIS) Project, MIT Sloan School of Management.

[35] Wang, Y. R. & Madnick, S. E. (1990). *A Source Tagging Theory for Heterogeneous Database Systems.* To Appear in the 11th International Conference on Information Systems. Copenhagen, Denmark. December 1990.

[36] Wiederhold, G. (1987). *KSYS: an architecture for integrating databases and knowledge bases.* Computer Science Department, Stanford University, Stanford, CA 94305. 1987.

[37] Yuan, Y. (1990). *The design and implementation of system P: A polygen database masnagement system.* (CIS-90-07) Composite Information Systems Laboratory, Sloan School of Management, MIT, Cambridge, MA. May 1990.

1-10-91